

### REMARKS

Claims 25, 27-31, 34-38, 40-41, and 43-47 are pending in this application. Claims 34 and 44 are amended herein to clarify that “the object machine is implemented as a function unit that ...processes the at least one hierarchical tree to a runtime system of the automated system” as well as correct an informal error. Claims 25, 28, 38, and 47 are further amended herein to correct informal errors. Claims 25, 27-31, 34-38, 40-41, and 43-47 are rejected as being anticipated by U.S. Published Patent Application No. 2006/0259157 to Lo et al. (“Lo”) under 35 U.S.C. 102(e). Reconsideration of the final rejection is respectfully requested in view of the following remarks.

In response to the final Office Action mailed December 11, 2007, Applicant respectfully requests that the Examiner amend the present application in the manner set forth in this Amendment. Applicant submits that this Amendment After Final Rejection places this application in condition for allowance by amending the claims in a manner that is believed to render the pending claims allowable over the cited art and/or at least place this application in better form for appeal. This Amendment was not presented earlier because Applicants believed that the prior responses placed this application in condition for allowance for at least the reasons discussed in the prior responses. Accordingly, entry of the present Amendment, as an earnest attempt to advance prosecution and/or to reduce the number of issues, is requested under 37 C.F.R. §1.116. In the event that the Examiner declines to enter the present Amendment, and (i) any portion of the present Amendment would place some of the claims in better form for appeal if a separate paper were filed containing only such amendments or (ii) any proposed amendment to any claim would render that claim allowable, Applicant respectfully requests that the Office inform Applicant of the same pursuant to MPEP §714.13.

#### **1. 35 U.S.C. 102(e) Rejections**

Claims 25, 27-31, 34-38, 40-41, and 43-47 are rejected as being anticipated by U.S. Published Patent Application No. 2006/0259157 to Lo et al. (“Lo”) under 35 U.S.C. 102(e). Per MPEP 2131, “[a] claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference.” *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). Applicant submits that independent Claim 25 is not anticipated by Lo as Lo does not expressly or inherently teach or suggest “converting the symbolic representation of the hierarchical tree to

physical addresses to generate a loadable program in the form of an executable program or operator tree.” The Examiner, however, maintains that paragraph [0082] of Lo teaches this limitation by disclosing that the configuration data comprises information relating to the mapping of logical to physical I/O. Applicants respectfully disagree with the Examiner’s position.

Lo is directed to providing a plurality of engineering tools at a centralized web server, having the user write programming code using the tools, and downloading the code to a programmable controller. In this way, a plurality of users may write programming code using the same version of software, for example, and may more easily work on collaborative projects. At paragraph [0048], Lo discloses engineering tools for generating programming code that reside on a web server and are capable of operating on a web-client in a browser application. A user accesses the web server over a network with a client device, i.e. personal computer, to access the engineering tool. The engineering tool is web-enabled and, in its most basic form, comprises an editor or editors that may be run on the web-browser by the user. According to Lo, at paragraph [0050], the transmission between the client and the server is preferably in the form of an XML document. Regarding any steps after generating the code, at paragraphs [0057] – [0062], Lo discloses that “[a]fter the code is created, debugged, and compiled, it is downloaded to a programmable controller...[I]n one embodiment programming a program controller comprises two steps: establishing communication between the controller and the server...and downloading the programming code to the controller over the network.”

At paragraph [0082] (the paragraph that the Examiner cites as disclosing the limitation of “converting the symbolic representation of the hierarchical tree to physical addresses to generate a loadable program in the form of an executable program or operator tree”), Lo states that the generating of program code may comprise the generation of configuration data. According to Lo, “[c]onfiguration data often comprise information such as...information relation to the mapping of logical to physical I/O.” The Examiner has maintained that the XML documents of Lo have the structure of hierarchical trees. *Assuming arguendo* for the purpose of this argument only that the XML documents are in the form of hierarchical trees, Lo does not teach or suggest “converting the symbolic representation of the hierarchical tree [(XML documents)] to physical addresses to generate a loadable program in the form of an executable program or operator tree” through its disclosure of the mapping of logical to physical I/O. Converting the symbolic

representation of XML documents to physical addresses is not the mapping of logical to physical I/O. In view of the above, Applicants submit that Claim 25, and all claims dependent therefrom, are in condition for allowance.

For the reasons set forth above with respect to Claim 25, Lo also does not teach or suggest the elements of Claim 38, namely “a component to convert the symbolic representation of the hierarchical tree to physical addresses to generate a loadable program.” Accordingly, Claim 38, and all claims dependent therefrom, are also in condition for allowance.

Moreover, Claim 47, requires a computer program implementing a method for executing a program for an industrial automation system, comprising:...“converting the symbolic representation of the hierarchical tree to physical addresses to generate a loadable program in the form of an executable program or operator tree.” For the reasons provided above with respect to Claim 25, Claim 47, and all claims dependent therefrom, are also in condition for allowance.

## **2. Dependent Claims**

The dependent claims provide further reason for allowance.

Applicant has amended Claims 34 and 44 to clarify that “the object machine is implemented as a function unit that is closed and that processes...at least one hierarchical tree to a runtime system of the automated system.” Claims 34 and 44 provide further reason for allowance because Lo is wholly silent as to an object machine that is implemented as a function unit that is closed and that processes at least one hierarchical tree to a runtime system of the automated system. The Examiner maintained that paragraphs [0047] and [0048] disclose “running in an application making them operating system independent.” Thus, the Examiner does not appear to have addressed whether Lo discloses “an object machine...that processes...at least one hierarchical tree to a runtime system of the automated system.” Moreover, Applicant submits that paragraphs [0047] and [0048] are wholly silent as to “object machine processes...at least one hierarchical tree...to a runtime system of the automated system.” Accordingly, Claims 34 and 44 are in condition for allowance as Lo does not teach or suggest the limitations of Claims 34 and 44.

Claims 37 and 46 require that the objects of the machine-independent program present as a hierarchical object or operator tree are assigned a collection of infrastructure services or infrastructure functions that access the objects via containers assigned to the objects such that an infrastructure service or an infrastructure function can be used by all the objects. As set forth in

paragraph [0033], by so doing, “[i]nfrastructure services or corresponding functions are accessed via the container 32 and such access is the same for all objects in the hierarchical tree.” The Examiner points to paragraph [0064 ] and Lo’s disclosure of ACTIVEX controls and contends that it was well-known at the time the invention was made that ACTIVEX is Microsoft technology used for developing reusable object oriented software components. Paragraph [0064] of Lo specifically discloses that “Microsoft...provides software for allowing applications running on its Internet Explorer® to access client side system resources by means of ACTIVEX controls embedded in the application.” This, according to paragraph [0064] of Lo, allows the application to be installed on the client’s device. However, even *assuming arguendo* for the purpose of this argument only that Lo includes containers and the applications referred to by Lo in paragraph [0064] include a hierarchical tree (i.e. an XML document), nothing in Lo discloses assigning the hierarchical tree (i.e. an XML document) a collection of infrastructure services or infrastructure functions that access objects via the containers (i.e. ACTIVEX controls). In view of the above, Claims 37 and 46 are in condition for allowance.

**3. Conclusion**

Accordingly, Applicants submit that all claims are in condition for allowance and request that a Notice of Allowance be issued. The commissioner is hereby authorized to charge any appropriate fees due in connection with this paper, including the fees specified in 37 C.F.R. §§ 1.16 (c), 1.17(a)(1) and 1.20(d), or credit any overpayments to Deposit Account No. 19-2179.

Respectfully submitted,

Dated: 2/8/08

By: 

John P. Musone  
Registration No. 44,961  
(407) 736-6449  
Siemens Corporation  
Intellectual Property Department  
170 Wood Avenue South  
Iselin, New Jersey 08830